A New Strategy in Fuzzy Inference Systems and in A.I.: The Selective Rules Activation (SRA) Algorithm

Horia-Nicolai Teodorescu[1]
Polytechnic University of Iasi, Center for Fuzzy Systems & A.I.
Copou 11, Iasi 6600, Romania

Takeshi Yamakawa
Fuzzy Logic Systems Institute
Iizuka, Fukuoka 820, Japan

Abstract - In both crisp and fuzzy inference machines, the degree of parallelism required to yield one complete elementary inference (i.e. an inference for one-node and one-output variable) in one processing step is defined as the dimension of the inference. It is shown that the complexity of the hardware, respectively the complexity of the computation can be much decreased by using a selective activation of the inference rules.

I. INTRODUCTION

Fuzzy inference systems are used in both fuzzy controllers and fuzzy expert systems. To increase the control quality, as well as the power of the expert systems, one needs hugh number of inference rules to be used. Consider for example an inference with 10 inputs, each input having 11 linguistic degrees (and the same number of membership functions), and only one output. To produce such an inference, according to the current way of inferring, one needs 11^{10} rule chips (each performing the inference for one rule). Obviously, such a hardware system can not be practically conceived now, because of the cost, dimensions a.s.o.

The algorithm discussed in this paper allows the building of what can be named 'hierarchical selective fuzzy systems'. It opens the way toward inference chips with 10,000 rules on a chip. Possibilities to extend the algorithm to other types of intelligent systems also exist.

As known, fuzzy control systems, as well as general-use fuzzy inference engines are built around some basic blocks, such as elementary rules inference blocks, defuzzifiers, truncation blocks and summing blocks. While the last category of elementary blocks are needed only as one per complete inference machine (for one-output machines at least), the elementary inference (rule) blocks are needed in a large number (tens, or more). The SRA algorithm helps reducing this number.

2. INPUT-SPACE TO RULES-SPACE MAPPING IN A FUZZY SYSTEM

The way a fuzzy system actually performs its inference / control

[1] Part of this paper was written during the stay of the first author, as a Visiting Professor, to FLSI, Iizuka, Japan.

tasks can be conveniently described by a mapping of the input space into the discrete space of rules. (A similar description, based on graphs in the discrete space of linguistic degrees, based on the corresponding linguistic system attached to a logic fuzzy system, is presented in reference [1]).

Consider for instance a two-input, one-output fuzzy system described by rules of the type:

(R(h,i)) :
If x is A_h and y is B_i,
Then z is C_k

For the system to perform the inference, it has to fire the appropriate rule R(h,i), that is, a mechanism inside the system has to map the two-dimensional real intervals corresponding to the membership functions to which the instant inputs actually belong, to the sub-set of corresponding functions. Indeed, this is the basic mechanism of rules activating. To simplify the presentation, consider that all membership functions, for both inputs, are non-vanishing on finite intervals, and that these intervals overlap only two-by-two. These properties of the input membership functions give rise to an activating mechanism that fires only four rules at a time.

Compare this with the situation of activating all the rules at the same time, as generally used in fuzzy inference machines. Even the well known way of fuzzy rules number minimizing by interpolation preserves an unnecessary amount of rules activated at a moment (in fact, this kind of minimizing does minimize the number of rule-blocks in the machine with the payoff of precision, and is not a 'true', sound minimizing).

As the number of input variables and the number of membership functions of the variables increase, the number of needed rules is quickly increasing in the case of usual manner of global rules firing. In general, if there are n input variables, x_1, x_2, ...x_n, having respectively q_1, q_2,..., q_n membership functions, the number of elementary rules to be activated are, in the classical case:

$$N = q_1 \times q_2 \times q_3 \times \ldots \times q_n.$$

For an n-input system, each input variable having q membership functions, the total number of rules is:

$$N = q^n.$$

This algorithm was first presented in **[2], [3], [4]**.

3. MAIN PROPERTIES OF THE RULES-FIRE MAPPINGS

Property 1: Consider a fuzzy logic inference system mapping a one-dimensional (1-D) space into another 1-D space, by rules in the form:

If x is Ai Y Then y is Bj. (Rij)

The rules are supposed to satisfy the univocity condition, in the sense: for any A_i, only B_j is assigned.

Let the input space (see Figure 3) satisfy the property that the maximal number of overlapping membership functions is *h*.

Let x represent the actual value of the crisp input. Then, at most *h* input membership functions are actually non-vanishing, and correspondingly the same maximum number of inferences are performed at any moment. Following, a maximum number of *h* rules have to be activated.

The proof is obvious.

Property 2: Consider a fuzzy logic inference system with multi-dimensional input space

$$X_1 \times X_2 \times \ldots \times X_n,$$

and a one-dimensional output space Y. The input spaces are supposed to satisfy the following condition:

In the Xj-th space, the maximum number of overlapping membership functions is q(j).

The system is described by the rules (denoted by R(.,.,...,.)):

$R(i_1, i_2, i_3, \ldots, i_n; j)$:

If x_1 is A_{i1} & x_2 is A_{i2} & ... & x_{in} is A_{in},

Then y is B_j

where A_{i1}, ..., A_{in} denote fuzzy subsets of the first, ..., n-th input space respectively, and B_j denotes a fuzzy sub-set of the output space. Below, the above rules are named elementary rules. The elementary rules are supposed to be univoque, i.e., for any tuple (i_1, .., i_n), the index j is unique. Then, performing the inference requires concomitant activation of the maximum of

$$Q = q_1 \times q_2 \times \ldots \times q_n \text{ rules.}$$

By definition, the maximal number of simultaneously activated elementary rules in an inference engine is named the *dimension of the system.*

Remark that the dimension δ of a fuzzy logic engine is generally much lower than the total number of rules describing the system (engine) at hand.

4. ALGORITHM FOR SELECTIVE RULES ACTIVATION

In this section, the rule-blocks are supposed to be universal, i.e., they can be set at any moment to perform any specific rule. (For example, if every of the inferred/control rules is implemented on one chip, and the chip has some pins to select the rule actually performed for a given input - i.e., pins to define the input-to-output mapping - than this chip stands for a 'universal rule-block'.)

The rules selective activation algorithm (presented for a one-dimensional input space case, i.e., for a single input variable case) is derived from the above discussion, as follows:

1. Determine the dimension δ of the system.
2. Split the input space into δ sub-spaces, each of them including input fuzzy 'overlapping' subsets.
3. To each of the above subsets, assign a rule block, to perform inference when the actual input value ranges in the corresponding subsets (i.e., when the input value belongs to the domain on which the corresponding membership functions do not vanish).
4. Use appropriate selection means (circuits) for the setting of the rule block, such as it actually performs the desired rules.

5. DISCUSSION AND CONCLUSIONS

The presented algorithm can be quite easily implemented in hardware, such as a rule chip able to perform a great number of rules (let us say: 10,000) is easily conceivable. The principles of such hardware is already described in [3], [4]. (A patent was also applied for a basic variant.) The algorithm was implemented and experimentally tested for a two-input, one-output fuzzy logic system.

The SRA algorithm is believed to open a new era of integration for analog fuzzy logic circuits, as well as for fuzzy expert systems chips.

APPENDIX
(by H.N. Teodorescu)

The use of the selective rules activation algorithm is possible, in different adapted variants, in classic (i.e., crisp) expert systems and other intelligent systems too.

For example, in an expert system, the dimension of the system can be re-defined as the product of the numbers of rules (thumb rules, or even functions defined on intervals) assigned to each of the inputs.

The SRA algorithm can also be extended, in an obvious way, to **multi-valued logic** systems.

Obviously, the SRA algorithm can be used to improve the fuzzy logic inference software.

Moreover, the algorithm can be extended to hierarchical systems, assigning a dimension to every level of the system.

REFERENCES

1. H.N. Teodorescu: An Introduction to Linguistic Systems. In: H.N. Teodorescu, E.P. Klement: An Introduction to Fuzzy Systems in Engineering.

2. H.N. Teodorescu, T. Yamakawa: Architecture of a fuzzy inference engine with minimal rule blocks and adaptive membership functions. In vol. *Fuzzy Systems. Proc. ISKIT'92*, Iizuka, 1992. pp. 79 - 84

3. H.N. Teodorescu, T. Yamakawa: Architectures for rule-chips number minimizing in fuzzy inference systems. In vol. *Proceedings of the 2nd Int. Conf. on Fuzzy Logic and Neural Networks*. Vol. 1., pp. 547 - 550 (Jono Printing Co., 1992, Iizuka, Japan)

4. H.N. Teodorescu: Minimization of rule-chips in fuzzy control and fuzzy inference systems. In vol. *Proceedings of the 2nd Int. Conf. on Fuzzy Logic and Neural Networks*. Vol. 1., pp. 441 - 444 (Jono Printing Co., 1992, Iizuka, Japan)